Introduction to Cocoa and Cocoa Touch Development with Xcode

Session 424

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

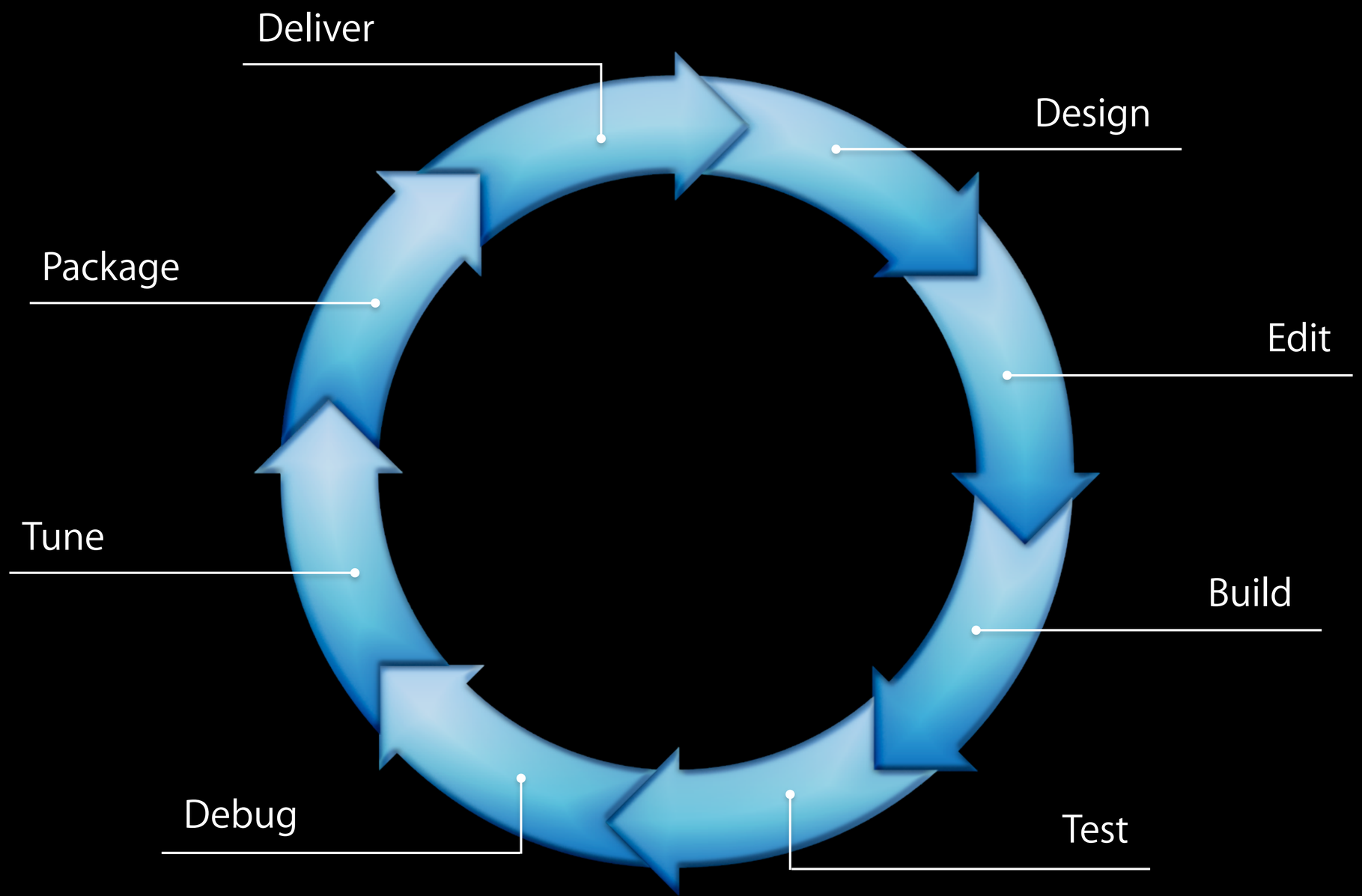
Introduction to Cocoa and Cocoa Touch Development with Xcode

Chris Espinosa
Manager, Xcode Core Team

Objectives

- Introduce you to the Apple developer tools suite
- Understand concepts and practices of developing with Xcode
- Walk through the basic development process
- Help you make the most of your week

The Big Picture



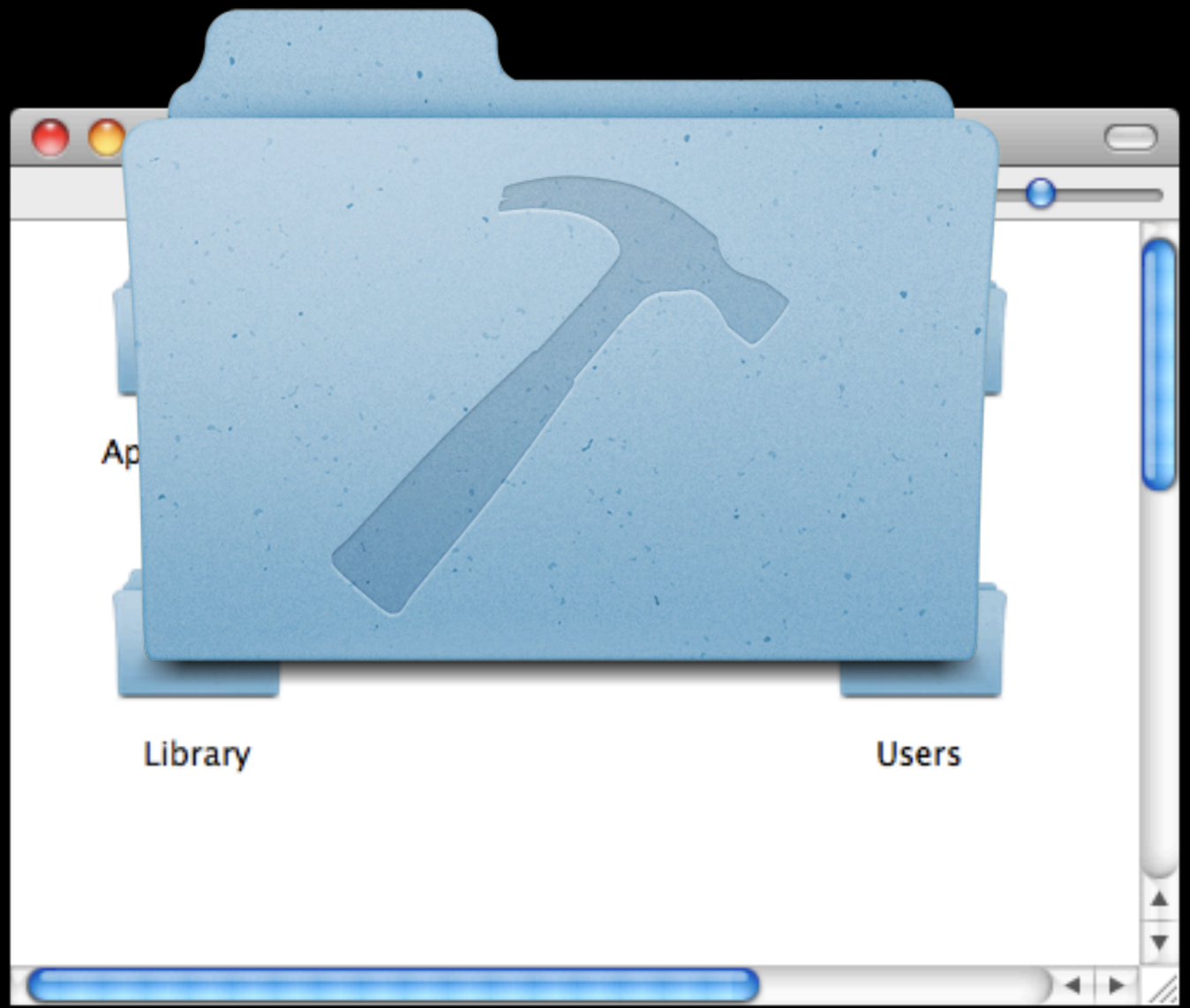
How to Get Xcode Tools

On your Mac OS X installation DVD

Downloaded as Xcode Tools from Mac Dev Center

Downloaded as iPhone SDK from iPhone Dev Center

The Developer Folder



Applications

7 items, 5.62 GB available

Xcode

Quartz Composer

Interface Builder

Instruments

Utilities

Graphics Tools

Performance Tools

Applications
1 of 7 selected, 5.62 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools

Applications
1 of 7 selected, 5.62 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools

Applications

1 of 7 selected, 5.62 GB available

Xcode

Quartz Composer

Interface Builder

Instruments

Utilities

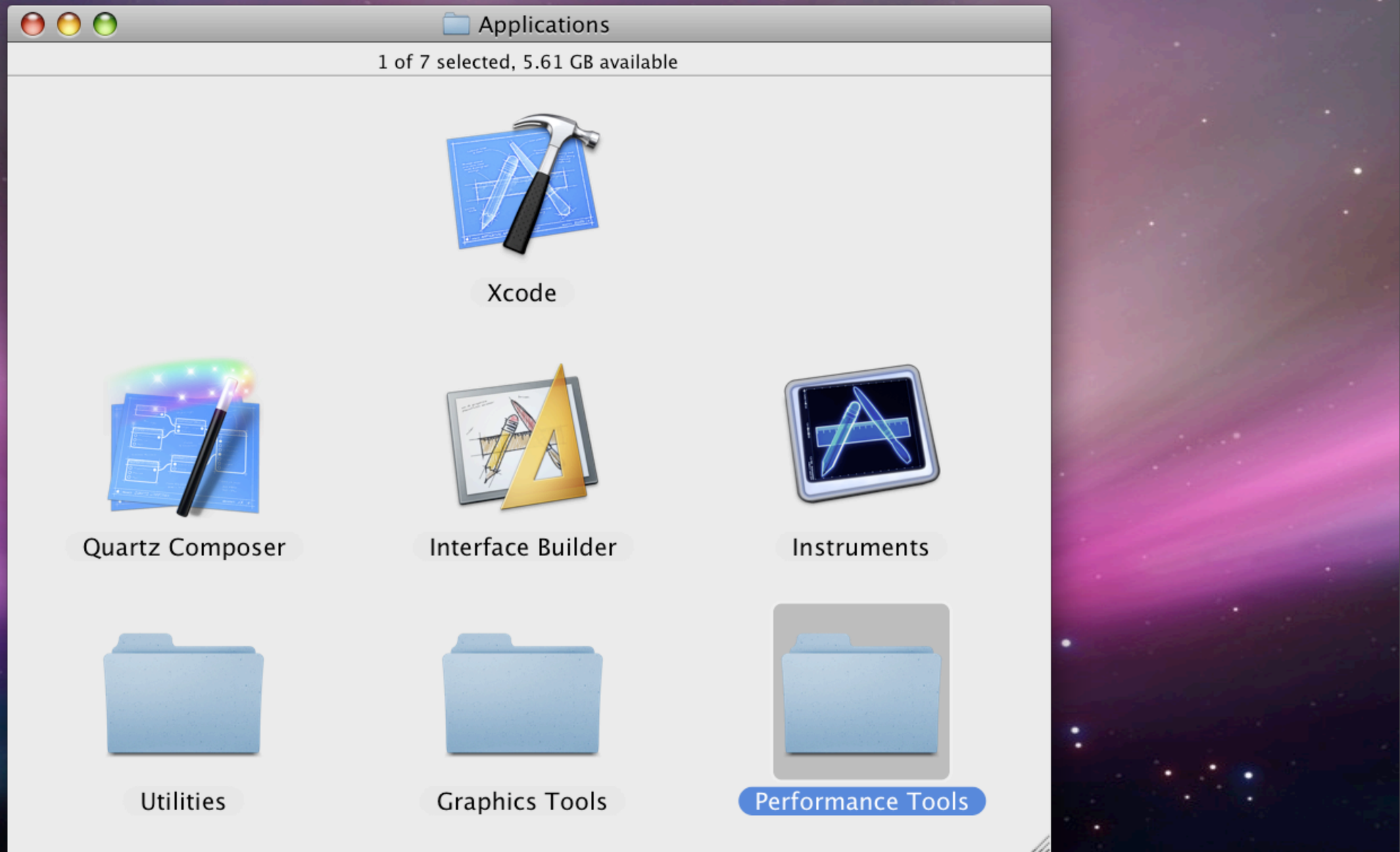
Graphics Tools

Performance Tools

Applications
1 of 7 selected, 5.62 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools

Applications
1 of 7 selected, 5.62 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools

Applications
1 of 7 selected, 5.61 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools


Applications
1 of 7 selected, 5.61 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools

Applications
7 items, 5.62 GB available
Xcode
Quartz Composer
Interface Builder
Instruments
Utilities
Graphics Tools
Performance Tools

Using Multiple Developer Folders

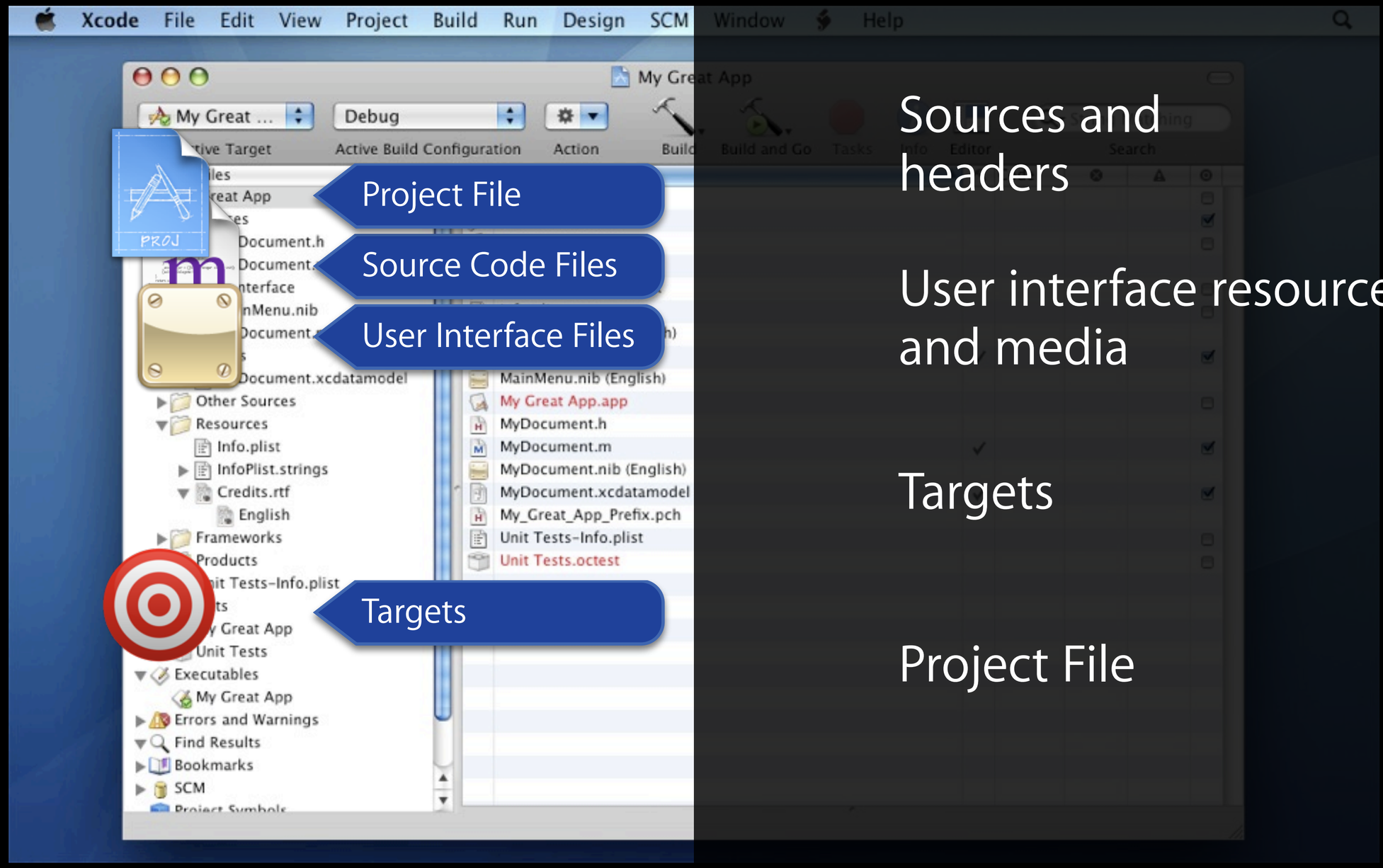
Developer

DeveloperBeta

Working with Xcode

What's an Xcode Project?

The set of things needed to build your software



Sources and headers

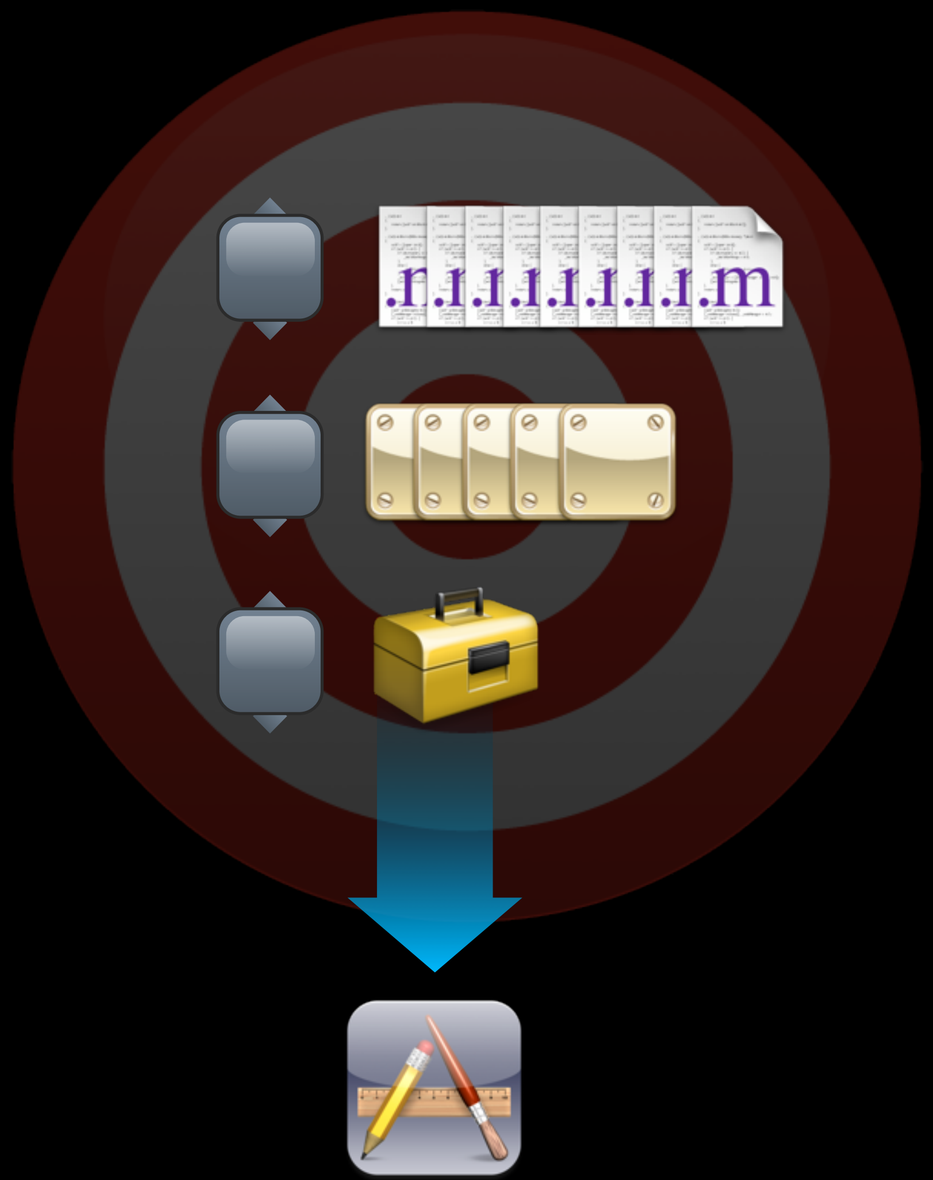
User interface resources and media

Targets

Project File

What's in a Project File?

- References to sources and headers
- Groups to organize files
- Targets to build products
- Executables to control running and debugging
- User settings
  - Bookmarks and history
  - Window layout
  - Breakpoints
  - User-specific configuration



What's in a Target?

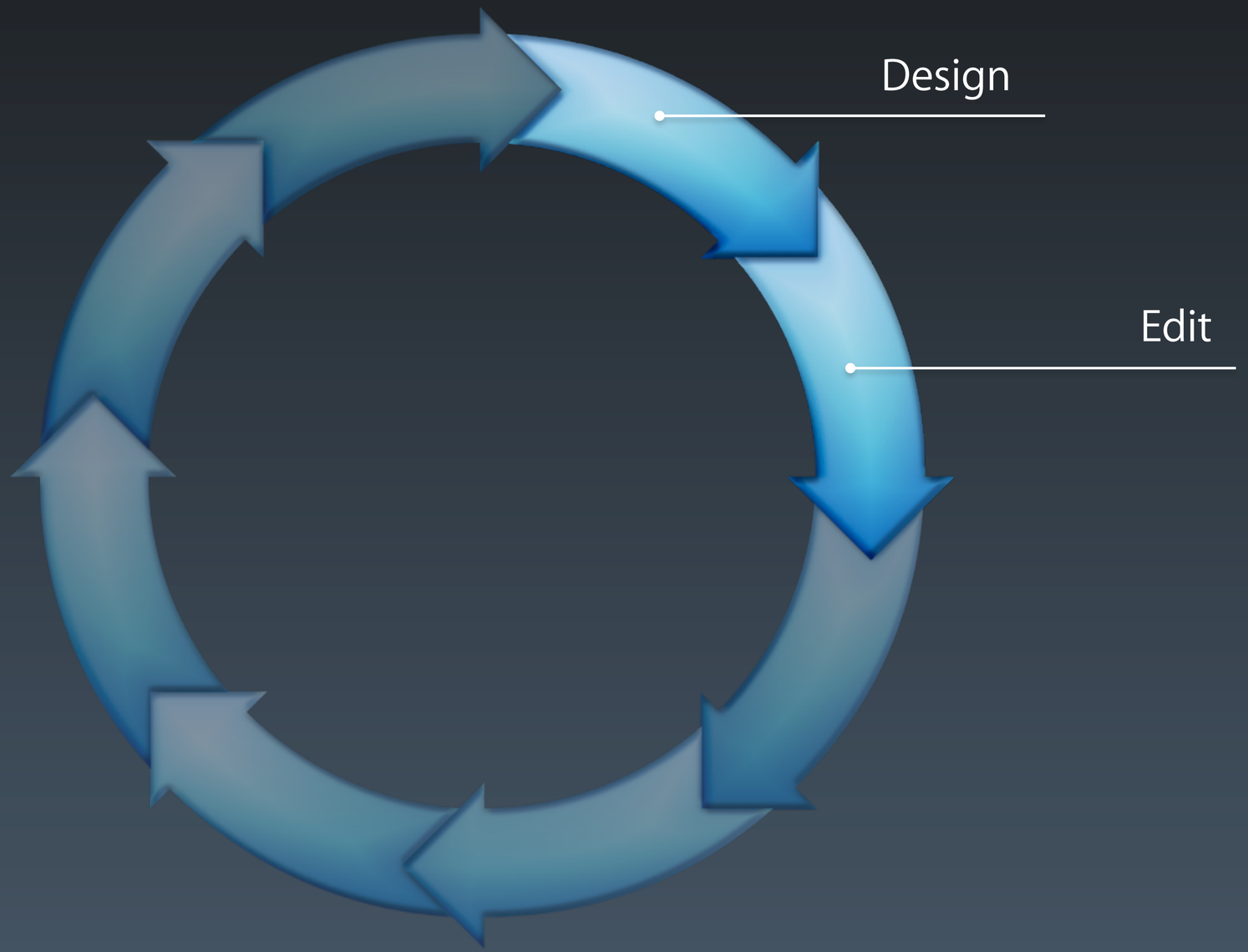
Instructions for building one product

- References some or all source files in the project
- Contains build phases—the high-level sequence of steps
- Build rules determine how to handle each file type
- Build settings control how it's done
- Can depend on one or more other targets

Demo

Rick Ballard
Xcode Engineer

Design and Edit



Text Editing

Customize to your working style

```
#import "AmeobaMine.h"
#import "Game.h"

@implementation AmeobaMine

- (void)setGeneration:(NSInteger)gen {
    generation = gen;
}

- (void)explode {
    if (generation > MAXAMEOBAGENERATION) {
        [super explode];
    } else {
        AmeobaMine *newMine = [[AmeobaMine alloc] initInGame:game];
        [newMine setHigh:highPoint low:lowPoint];
        [newMine setVelocity:NSMakeSize(-vel.width, -vel.height)];
        [newMine setLocation:NSMakePoint(pos.origin.x, pos.origin.y)];
        [newMine setAcceleration:NSMakeSize(0.0, 0.0)];
        [self setGeneration:generation + 1];
        [newMine setGeneration:generation + 1];
        [game addGamePiece:newMine];
    }
}

@end
```

Text Editing

Syntax coloring, fonts, and themes

```
#import "AmeobaMine.h"
#import "Game.h"

@implementation AmeobaMine

- (void)setGeneration:(NSInteger)gen {
    generation = gen;
}

- (void)explode {
    if (generation > MAXAMEOBAGENERATION) {
        [super explode];
    } else {
        AmeobaMine *newMine = [[AmeobaMine alloc] initInGame:game];
        [newMine setHigh:highPoint low:lowPoint];
        [newMine setVelocity:NSMakeSize(-vel.width, -vel.height)];
        [newMine setLocation:NSMakePoint(pos.origin.x, pos.origin.y)];
        [newMine setAcceleration:NSMakeSize(0.0, 0.0)];
        [self setGeneration:generation + 1];
        [newMine setGeneration:generation + 1];
        [game addGamePiece:newMine];
    }
}

@end
```

Text Editing

Spacing, tabs, and indentation preferences, manual and automatic

```
#import "AmeobaMine.h"
#import "Game.h"

@implementation AmeobaMine

- (void)setGeneration:(NSInteger)gen {
        generation = gen;
}

- (void)explode {
        if (generation > MAXAMEOBAGENERATION) {
                [super explode];
        } else {
                AmeobaMine *newMine = [[AmeobaMine alloc] initInGame:game];
                [newMine setHigh:highPoint low:lowPoint];
                [newMine setVelocity:NSMakeSize(-vel.width, -vel.height)];
                [newMine setLocation:NSMakePoint(pos.origin.x, pos.origin.y)
                [newMine setAcceleration:NSMakeSize(0.0, 0.0)];
                [self setGeneration:generation + 1];
                [newMine setGeneration:generation + 1];
                [game addGamePiece:newMine];
        }
}

@end
```

Text Editing

Wrapping or nonwrapping

```
#import "AmeobaMine.h"
#import "Game.h"

@implementation AmeobaMine

- (void)setGeneration:(NSInteger)gen {
        generation = gen;
}

- (void)explode {
        if (generation > MAXAMEOBAGENERATION) {
                [super explode];
        } else {
                AmeobaMine *newMine = [[AmeobaMine alloc] initInGame:game];
                [newMine setHigh:highPoint low:lowPoint];
                [newMine setVelocity:NSMakeSize(-vel.width, -vel.height)];
                [newMine setLocation:NSMakePoint(pos.origin.x, pos.origin.y
                    )];
                [newMine setAcceleration:NSMakeSize(0.0, 0.0)];
                [self setGeneration:generation + 1];
                [newMine setGeneration:generation + 1];
                [game addGamePiece:newMine];
        }
}

@end
```

Text Editing

Line numbers, line number counter, column counter

AmeobaMine.m:18:17 -explode

```
#import "AmeobaMine.h"
#import "Game.h"

@implementation AmeobaMine

- (void)setGeneration:(NSInteger)gen {
        generation = gen;
}

- (void)explode {
        if (generation > MAXAMEOBAGENERATION) {
                [super explode];
        } else {
                AmeobaMine *newMine = [[AmeobaMine alloc] initInGame:game];
                [newMine setHigh:highPoint low:lowPoint];
                [newMine setVelocity:NSMakeSize(-vel.width, -vel.height)];
                [newMine setLocation:NSMakePoint(pos.origin.x, pos.origin.y)];
                [newMine setAcceleration:NSMakeSize(0.0, 0.0)];
                [self setGeneration:generation + 1];
                [newMine setGeneration:generation + 1];
                [game addGamePiece:newMine];
        }
}

@end
```

Text Editing

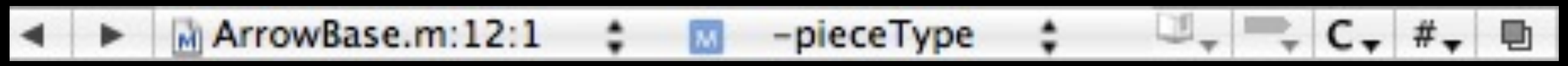
Code focus and focus follows selection

```
#import "AmeobaMine.h"
#import "Game.h"

@implementation AmeobaMine

- (void)setGeneration:(NSInteger)gen {
        generation = gen;
}

- (void)explode {
        if (generation > MAXAMEOBAGENERATION) {
                [super explode];
        } else {
                AmeobaMine *newMine = [[AmeobaMine alloc] initInGame:game];
                [newMine setHigh:highPoint low:lowPoint];
                [newMine setVelocity:NSMakeSize(-vel.width, -vel.height)];
                [newMine setLocation:NSMakePoint(pos.origin.x, pos.origin.y)
                [newMine setAcceleration:NSMakeSize(0.0, 0.0)];
                [self setGeneration:generation + 1];
                [newMine setGeneration:generation + 1];
                [game addGamePiece:newMine];
        }
}

@end
```

Navigation

Getting around

- Navigation bar and function pop-up



- Open Quickly
  - Navigate to project or framework file by name
  - Jump to declaration or definition of any symbol
- Command-Double-Click to jump to definition
- Class Browser

Code Sense

Xcode's code completion

- Automatically suggests completions in-line as you type
- Puts placeholder tokens in your code for arguments
- Key equivalents let you code fast

control > .	Complete now
control < ,	Show completion list
control ? /	Go to next placeholder

Edit All in Scope

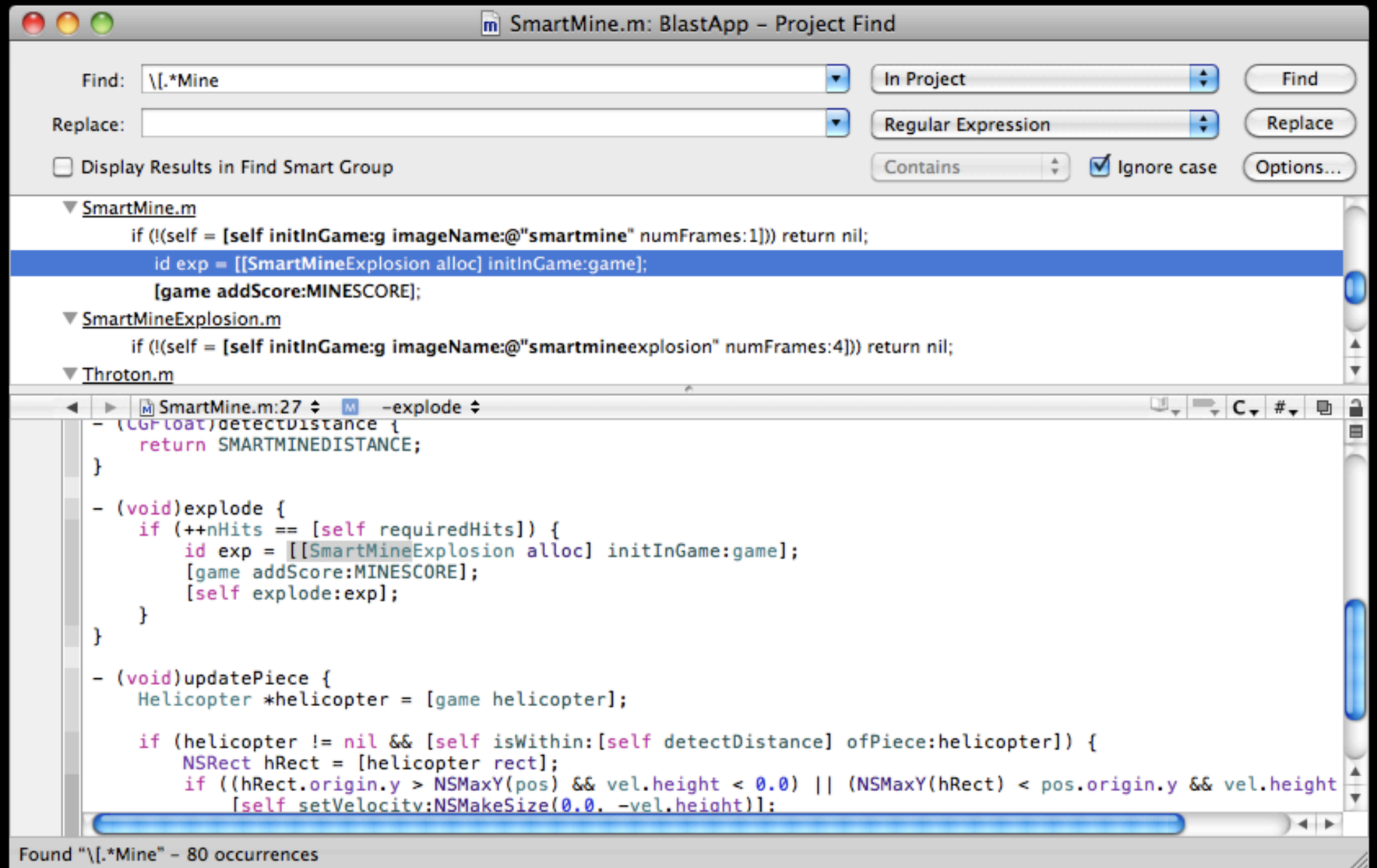
Fast local changes

- Make simple, common changes fast…
- …without even taking your hands off the keyboard!

⌘ command + control + T	Edit all in scope

Find and Replace

Fast projectwide searching



Refactoring

Semantic operations on your sources

BlastApp – Refactoring

Move Up — **explode** to Class SmallMine

Move up related methods

Snapshot ? Cancel Apply

▼Errors & Warnings

- Reference to instance variable 'generation' in method 'explode' won't be in scope if method is moved up. — AmeobaMine.m
- Reference to instance variable 'generation' in method 'explode' won't be in scope if method is moved up. — AmeobaMine.m
- Reference to instance variable 'generation' in method 'explode' won't be in scope if method is moved up. — AmeobaMine.m
- Moving method to class 'SmallMine' may cause message 'setGeneration:' to fail because the method is only declared in class 'AmeobaMine'. — AmeobaMine.m

▼Files

- AmeobaMine.m — 2 changes
- SmallMine.m — 1 change

@implementation SmallMine

```
 <codex/>
 */

#import "SmallMine.h"
#import "SmallMineExplosion.h"
#import "Game.h"

@implementation SmallMine
- (id)initInGame:(Game *)g {
    if (!(self = [self initInGame:g imageName:@"smallmine" numI
    [self setPerFrameTime:100000000];
    return self;
}

- (void)explode {
    id exp = [[SmallMineExplosion alloc] initInGame:game];
    [game addScore:MINESCORE];
    [self explode:exp];
}
@end
```

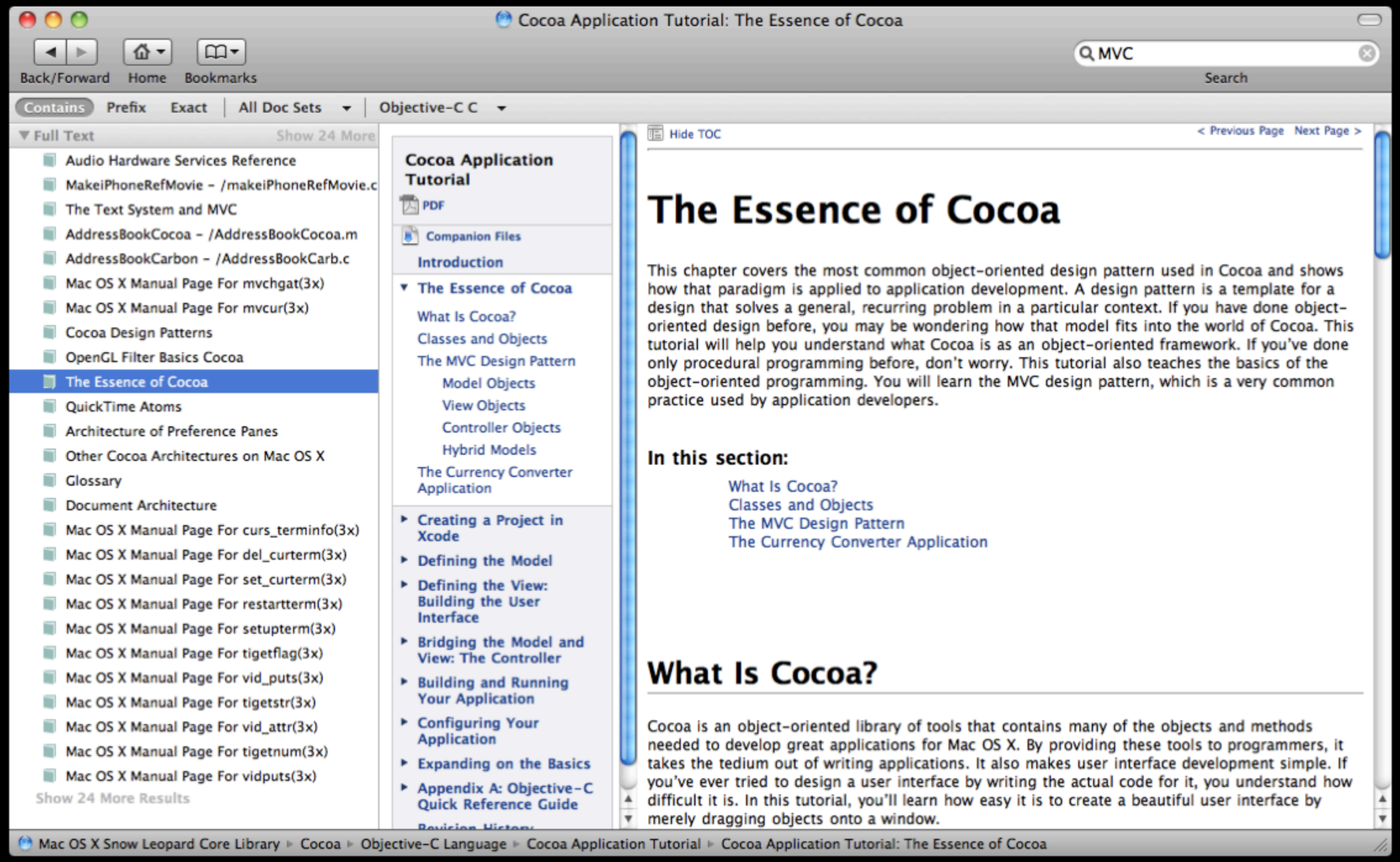
1

-explode

```
- (void)explode {
    id exp = [[SmallMineExplosion alloc] initInGame:game];
    [game addScore:MINESCORE];
    [self explode:exp];
}
- (void)explode {
        if (generation > MAXAMEOBAGENERATION) {
                [super explode];
        } else {
                AmeobaMine *newMine = [[AmeobaMine alloc] init
                [newMine setHigh:highPoint low:lowPoint];
                [newMine setVelocity:NSMakeSize(-vel.width, -v
                [newMine setLocation:NSMakePoint(pos.origin.x,
                [newMine setAcceleration:NSMakeSize(0.0, 0.0)]
                [self setGeneration:generation + 1];
                [newMine setGeneration:generation + 1];
                [game addGamePiece:newMine];
        }
}
@end
```

Getting Help

Developer Documentation

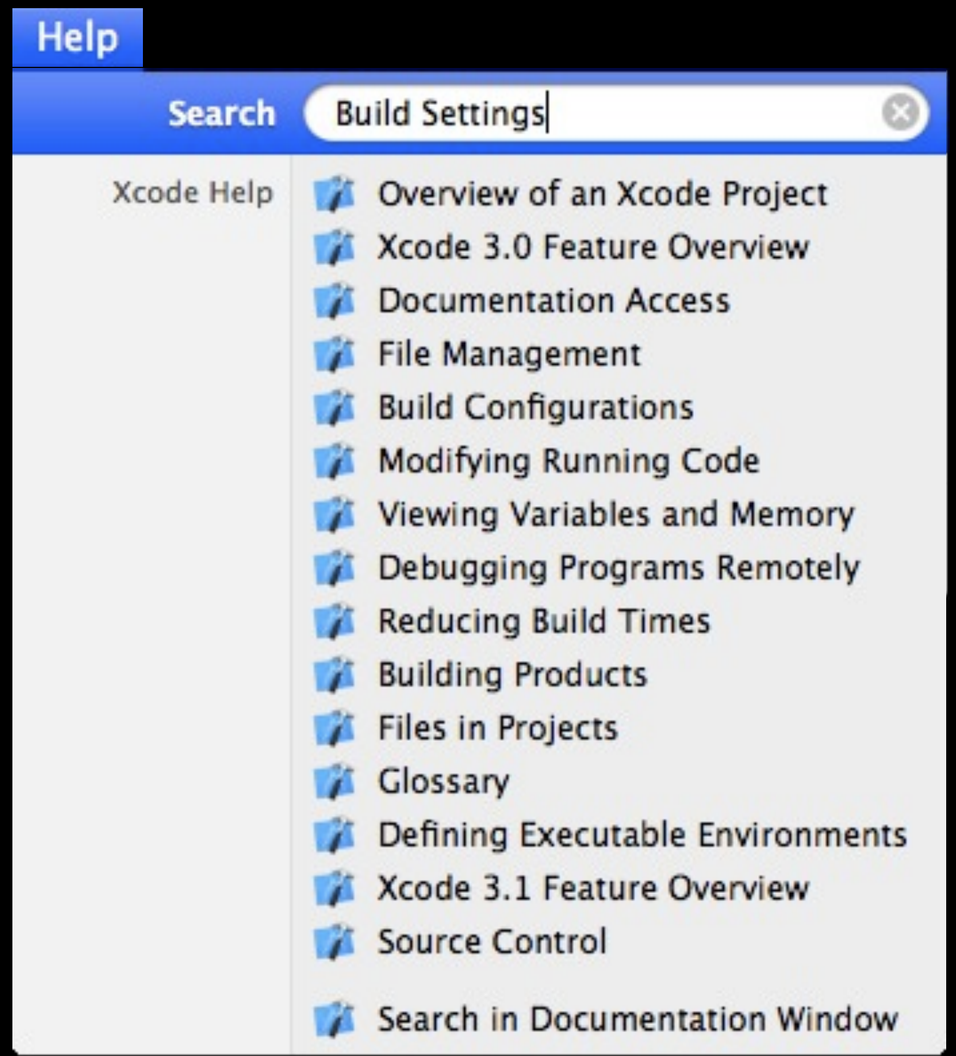


Getting Help

Developer Documentation

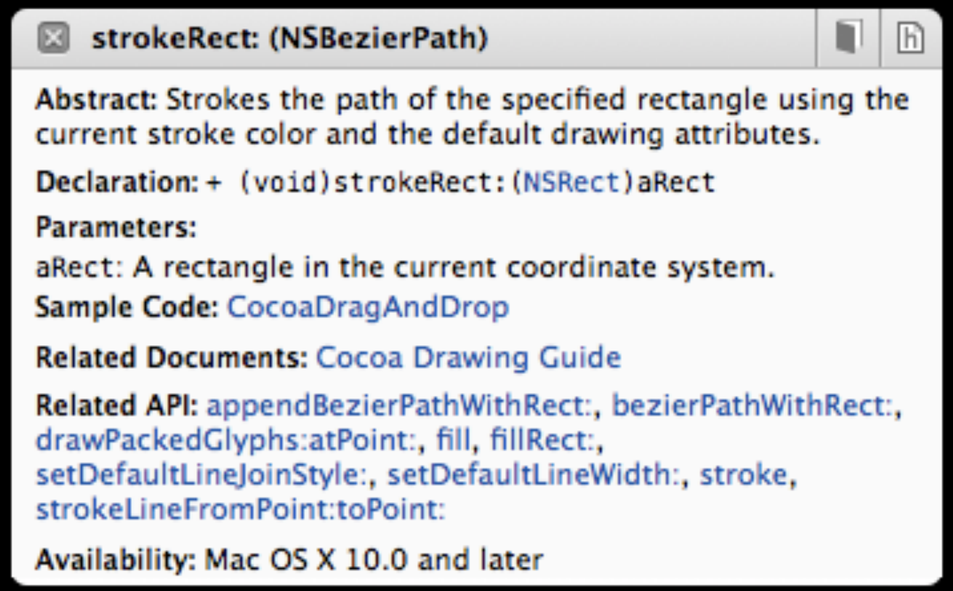
- Multiple full and comprehensive document sets
  - Xcode
  - Mac OS X
  - iPhone
- One-step searching of all documentation
- Super-fast searching of hundreds of megabytes of documents
- Regularly and automatically updated

Help Menu



- Fast access to Developer Documentation and Help
- Opens UNIX man pages directly

Quick Help



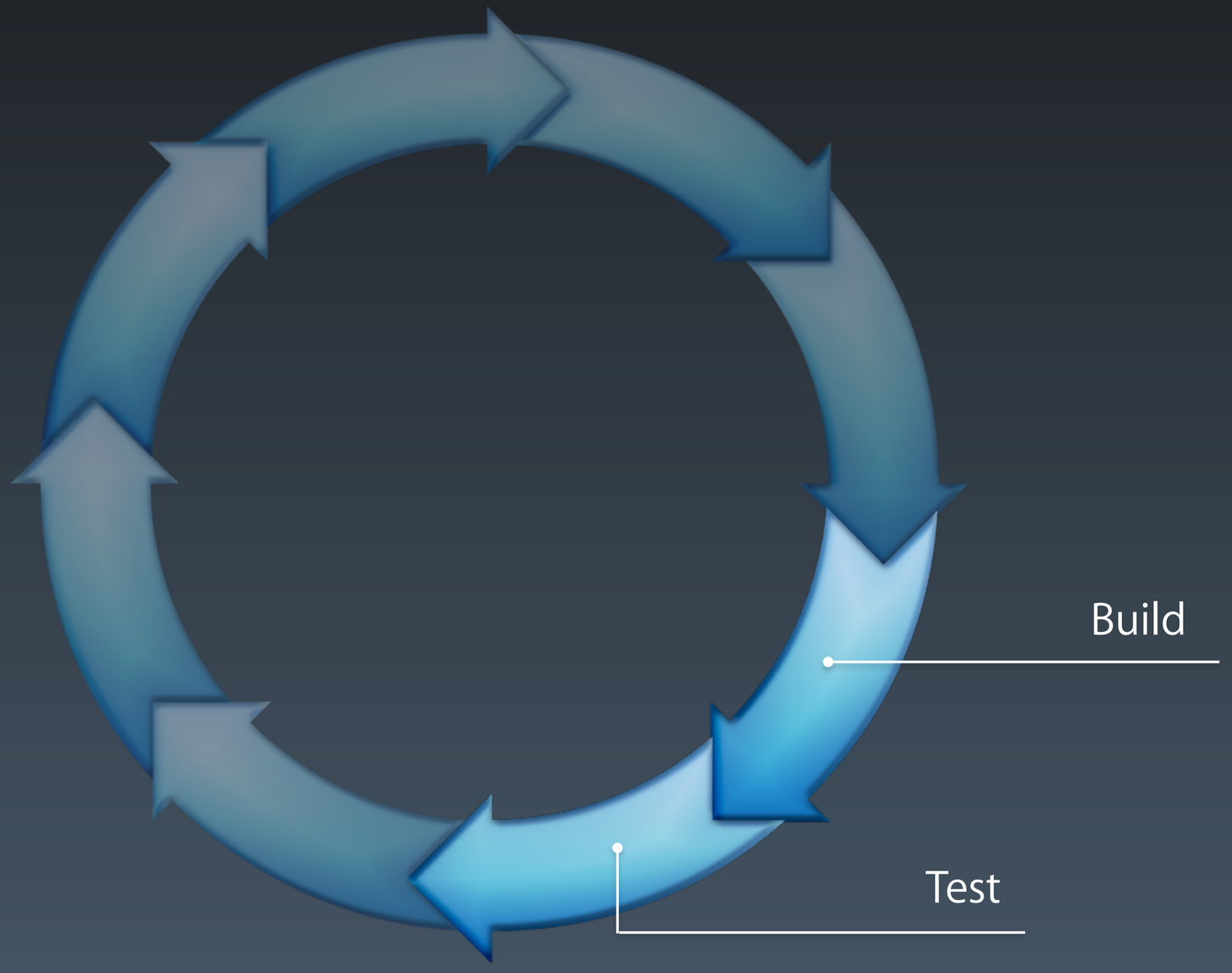
- API documentation in the editor
- Appears on use, or you can make it float
- Jump directly to:
  - Documentation page
  - Header file
  - Sample code

Demo

Text editing and getting help

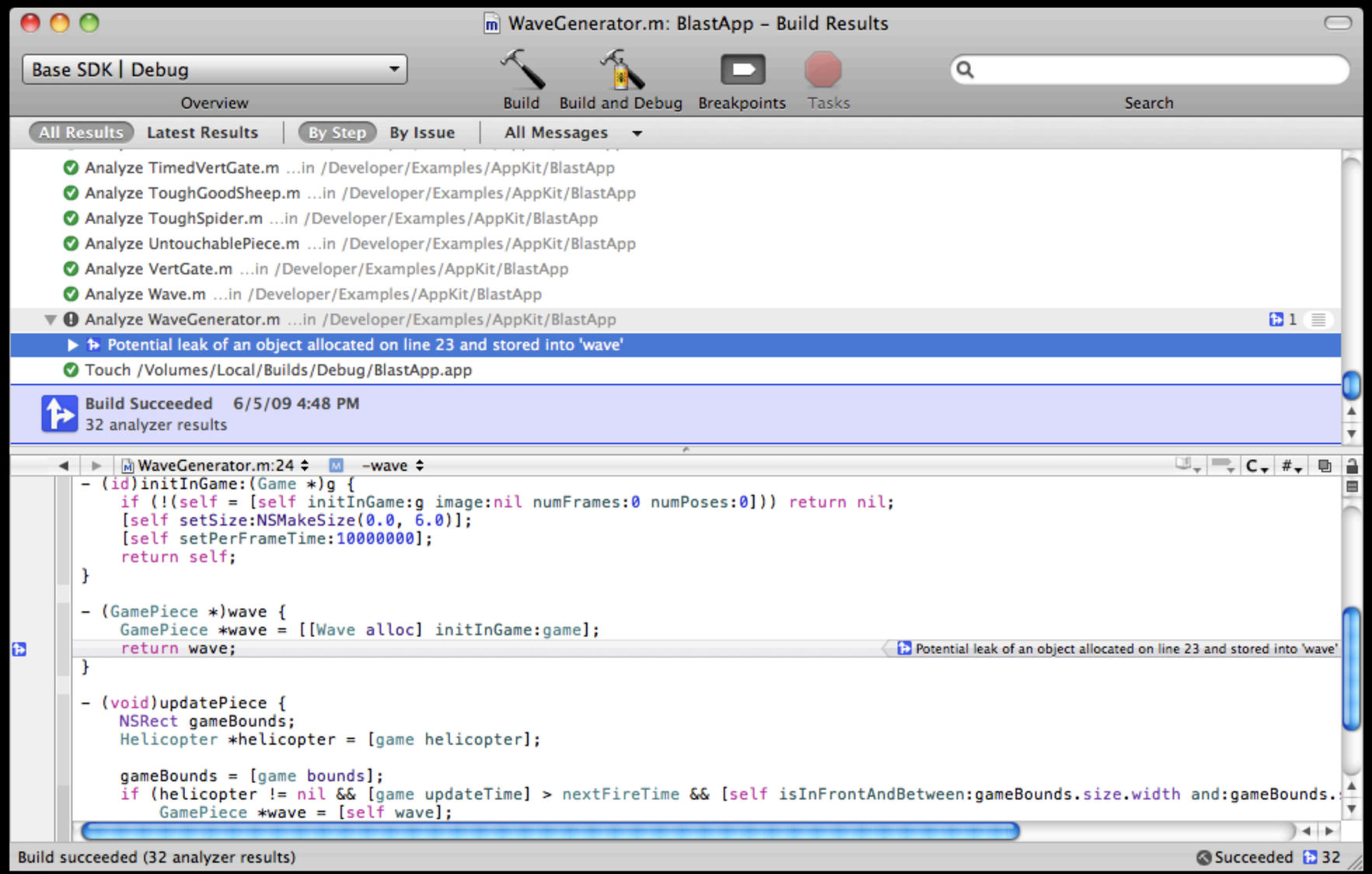
Brent Shank
Developer Tools Engineer

Build and Test

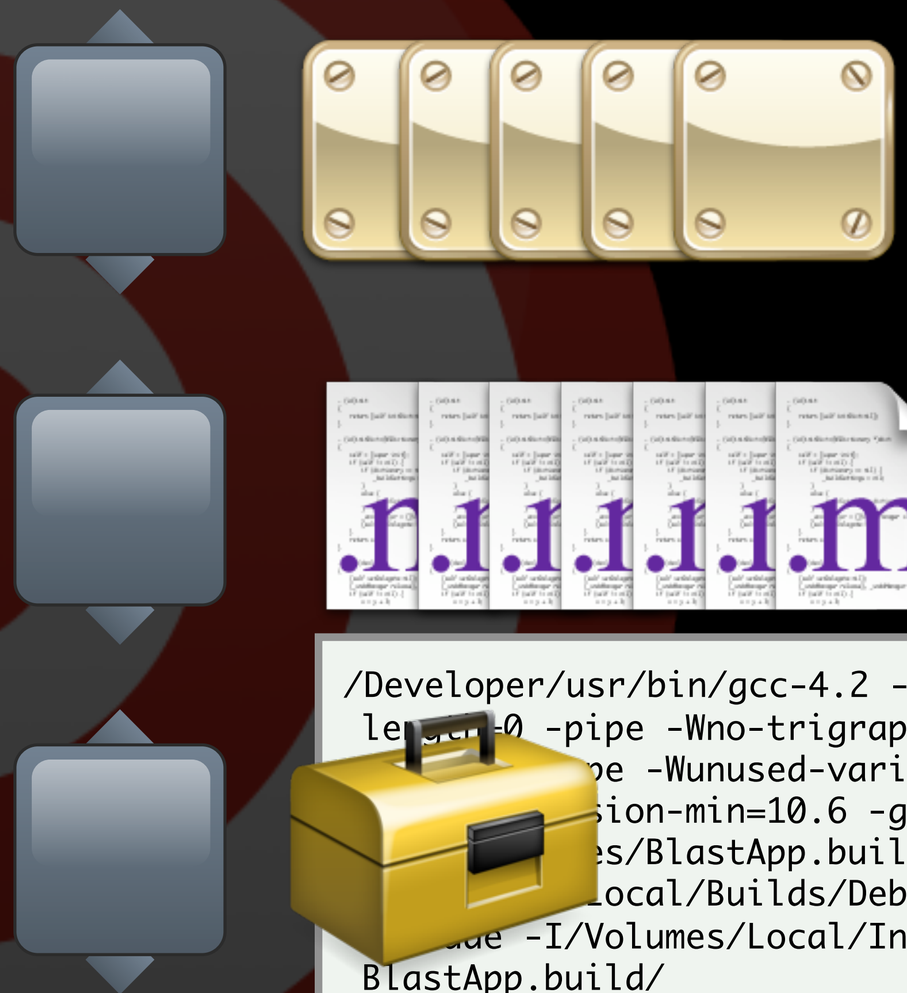


Building

Compiler results appear right in your editor



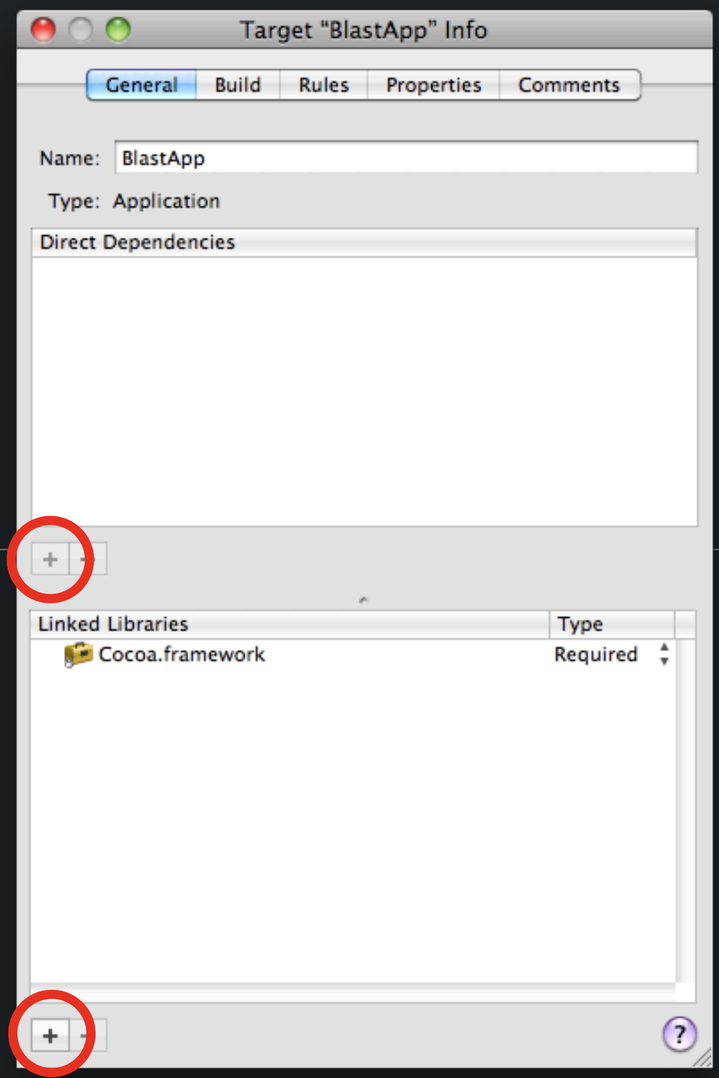
The Build Process



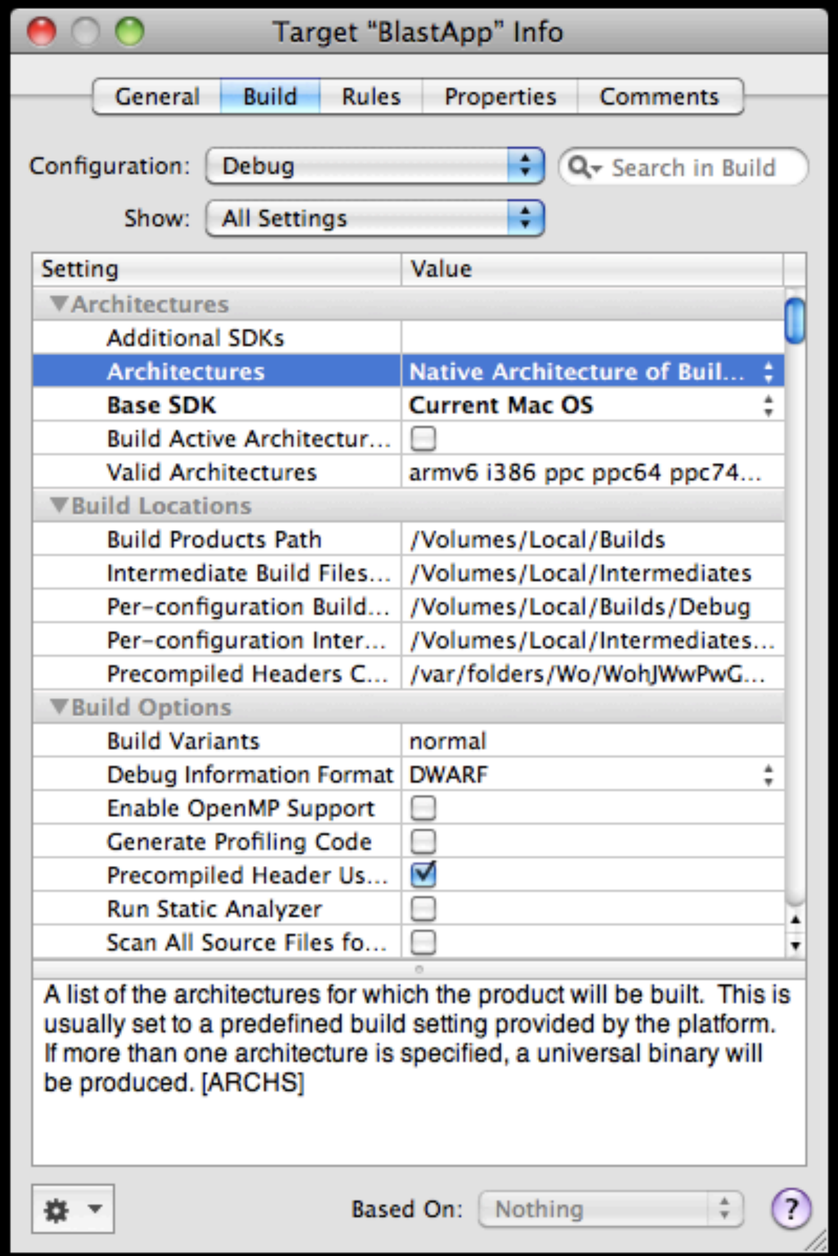
Tasks with Targets

Make one target depend on another

Use a framework or library in a target



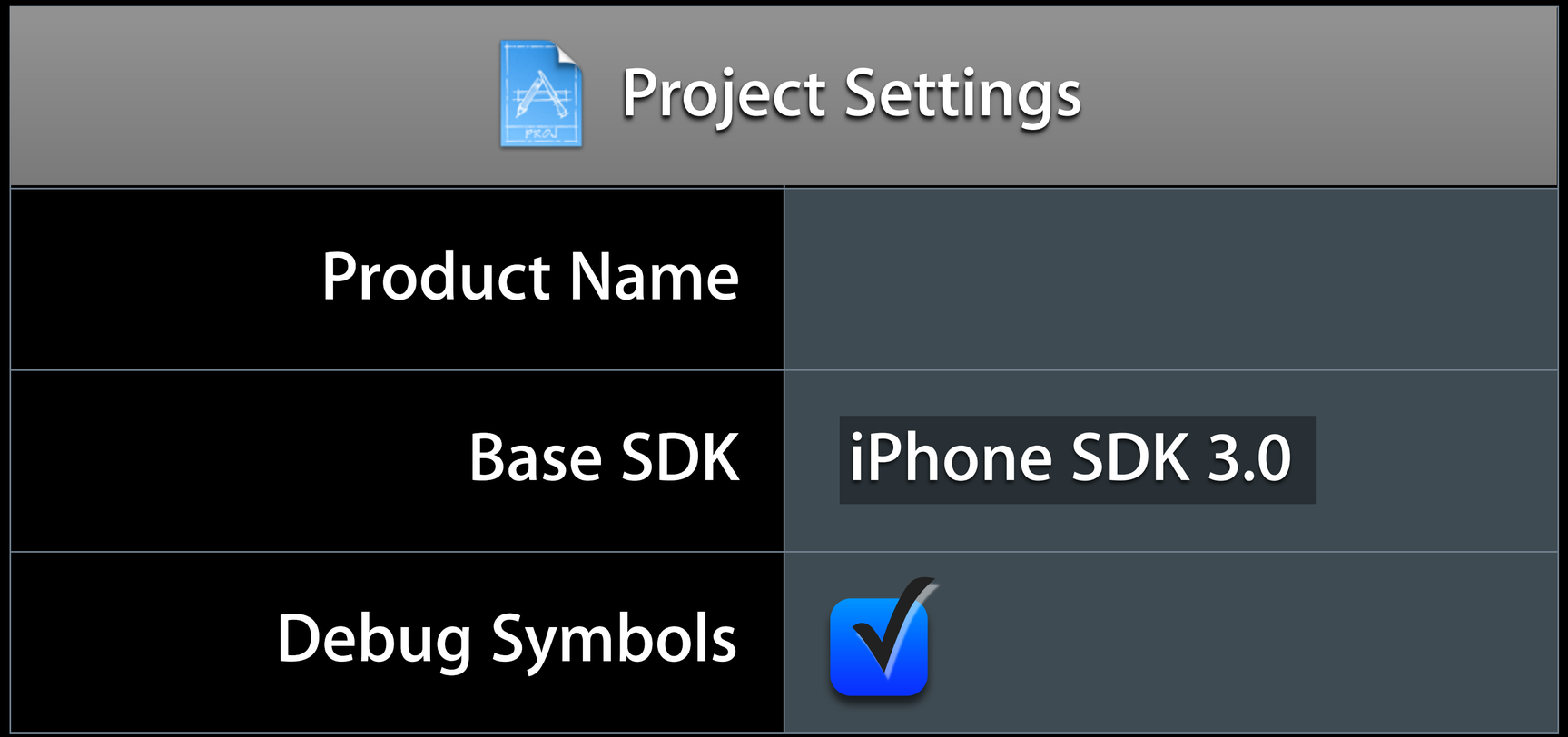
Build Settings



- Specific values for tool options
- Organized into configurations—debug, release, and so on
- Default values for everything—user can customize
- Arranged into inheritance hierarchy

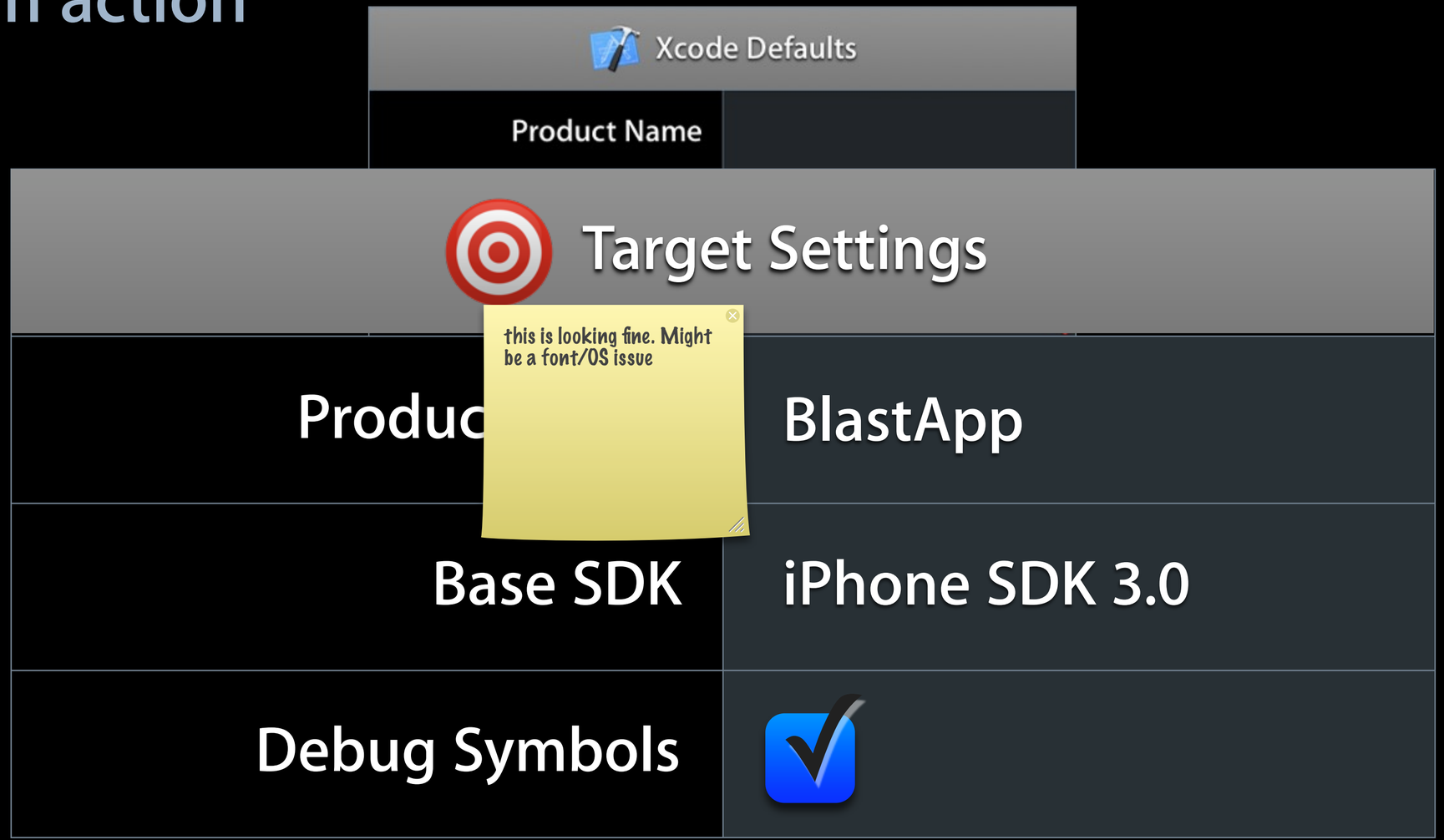
Build Settings

Hierarchy in action



Build Settings

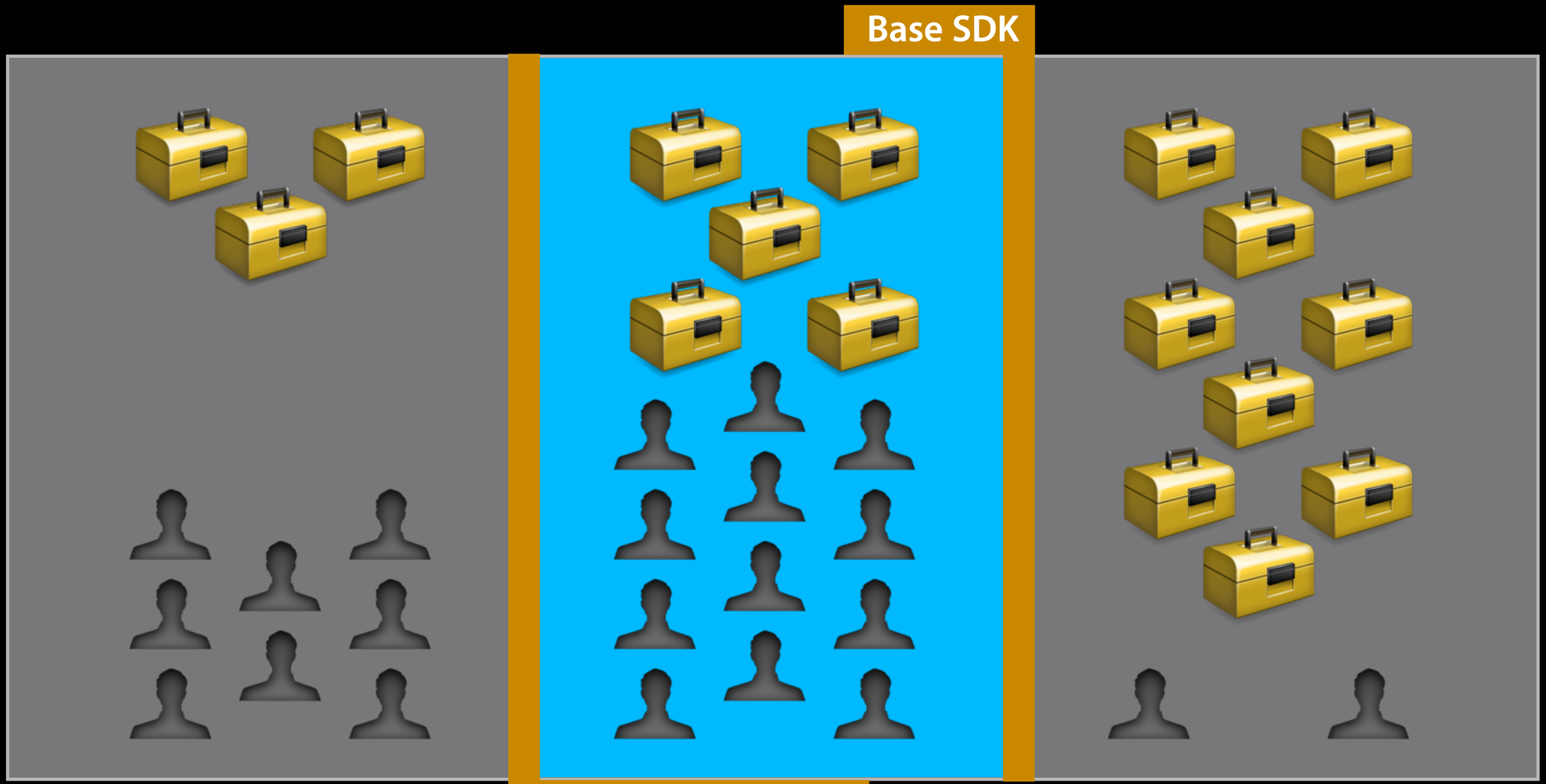
Hierarchy in action



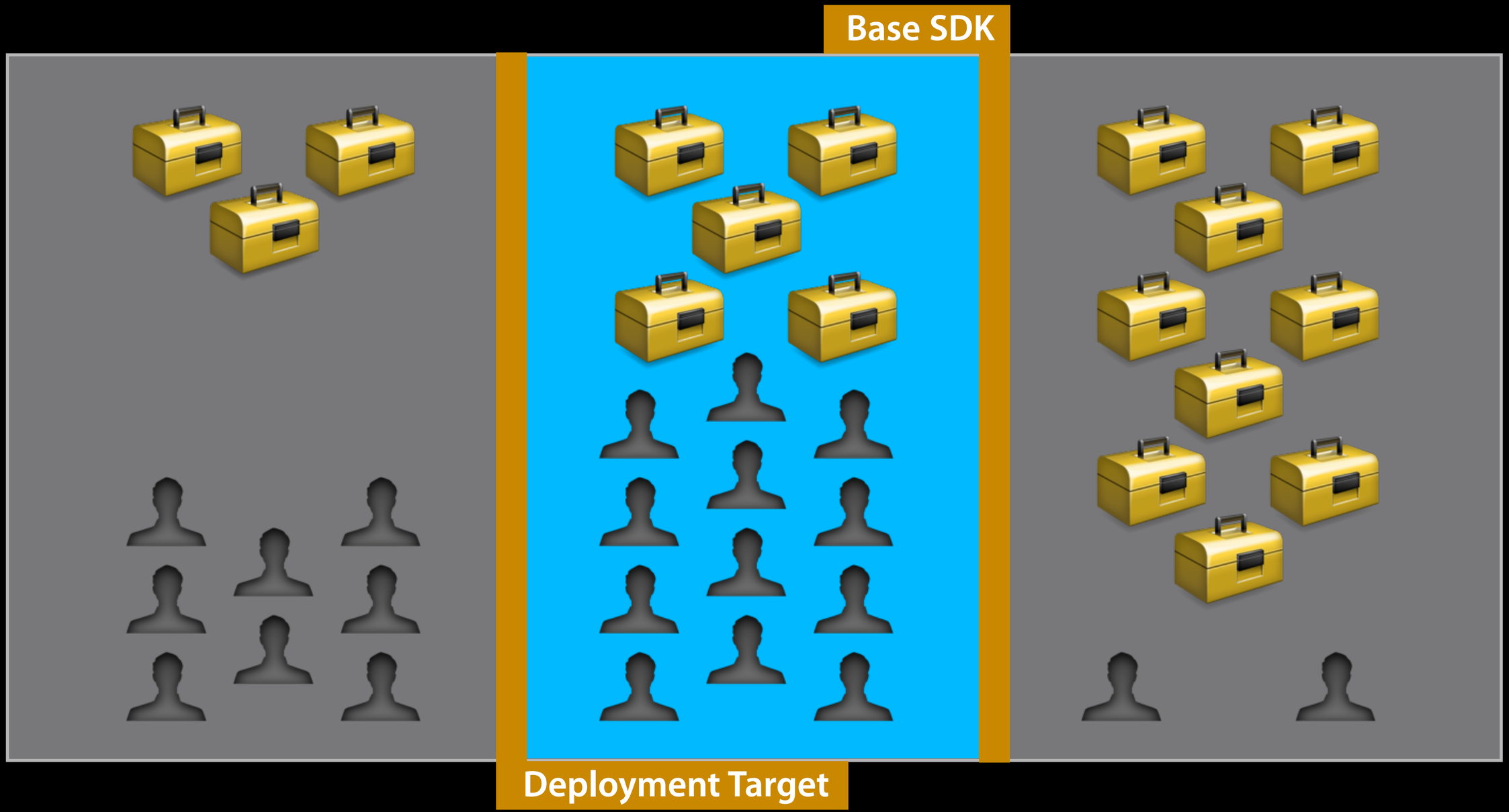
Tasks with Build Settings

Set the Build Product Name	Product Name
Find headers outside the project	Header Search Paths
Relocate the build folder	Build Products Path
Choose an API for your source	Base SDK
Set the earliest OS version	Deployment Target

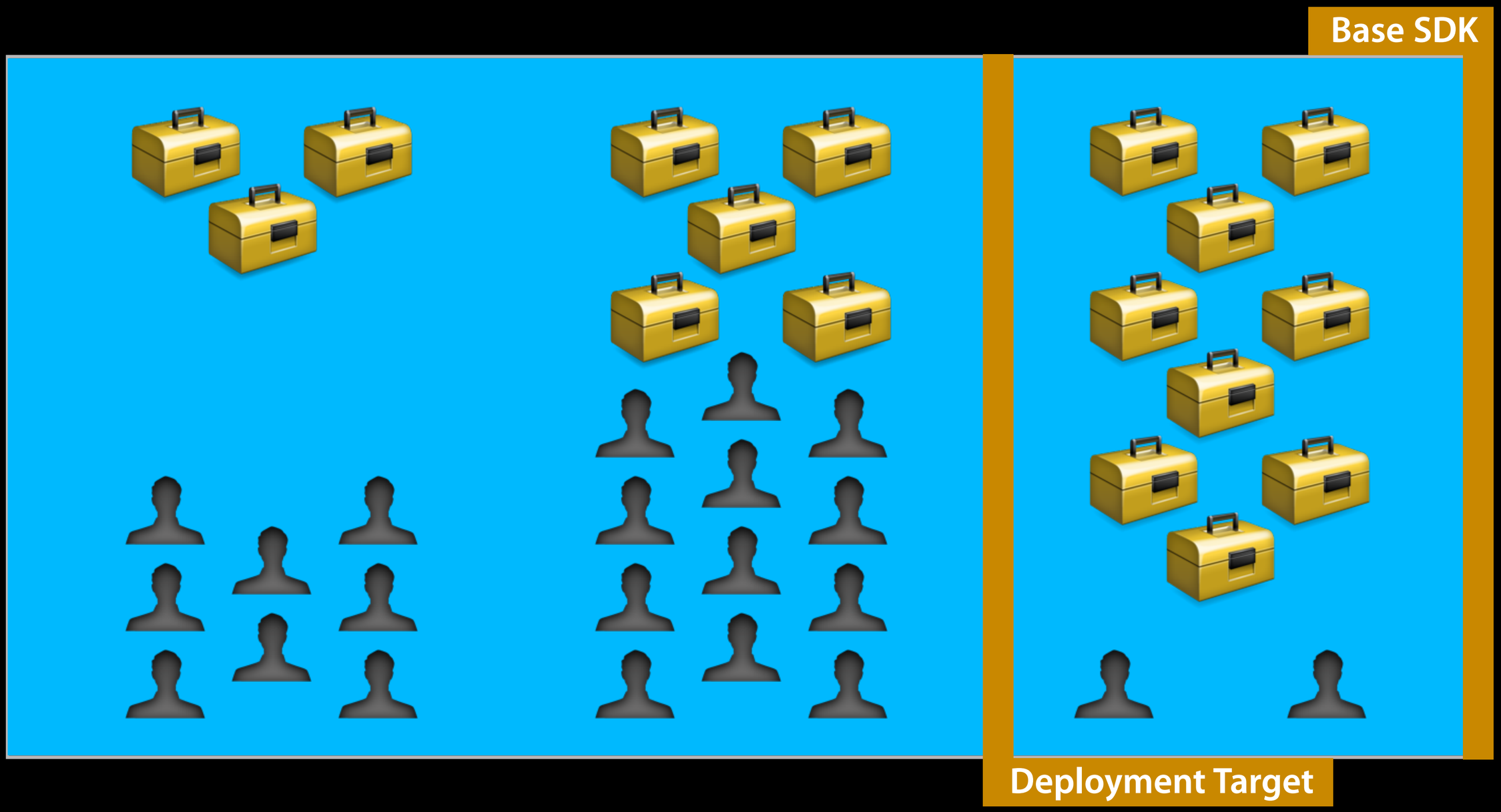
Balancing Features and Reach



Balancing Features and Reach

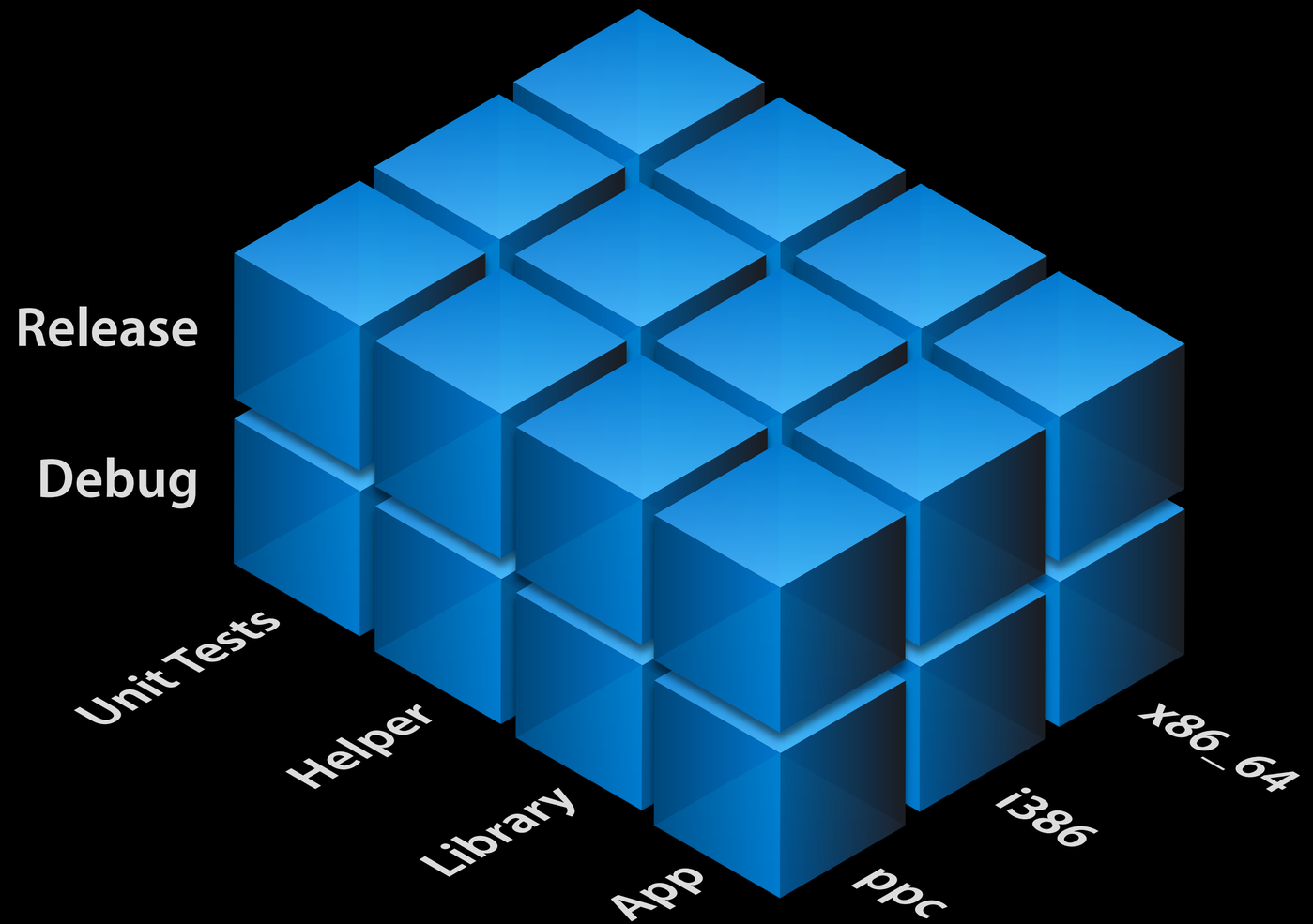


Balancing Features and Reach



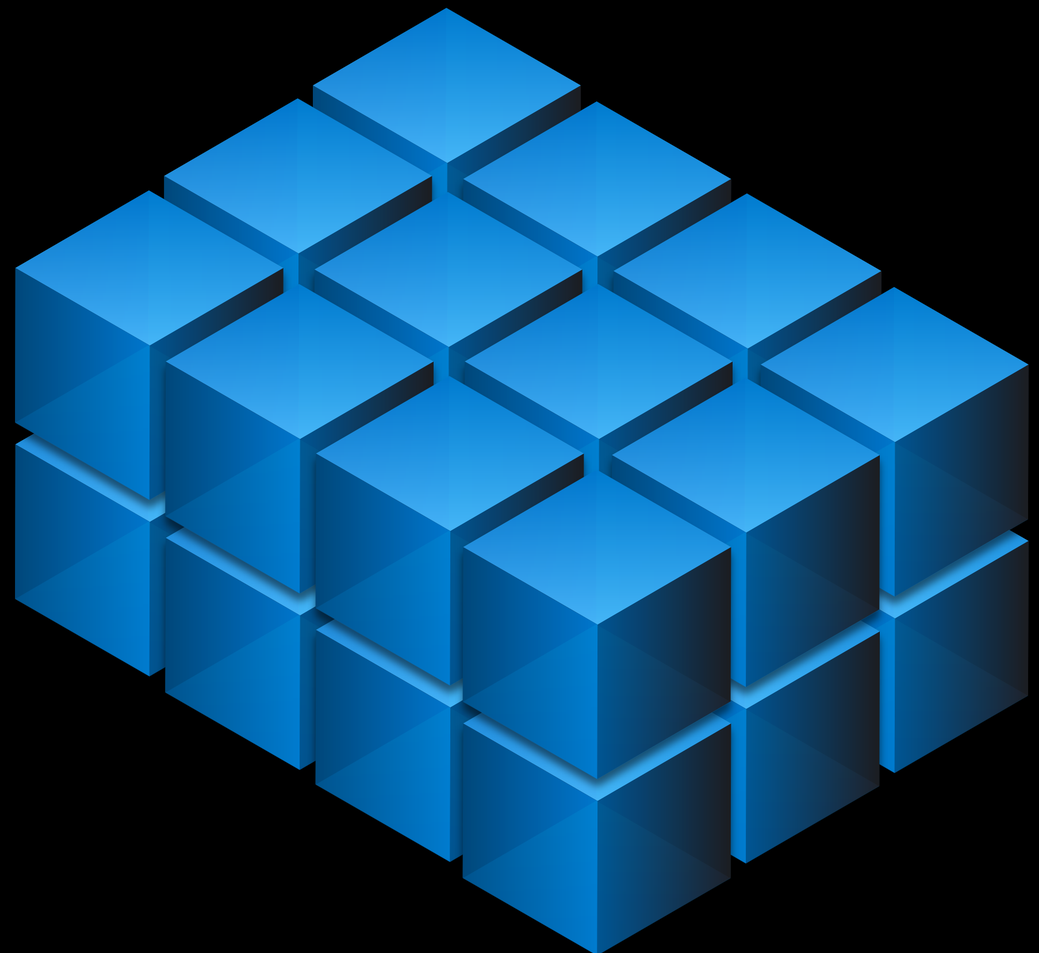
The State of Your Project

“Active” menu items and toolbar popups



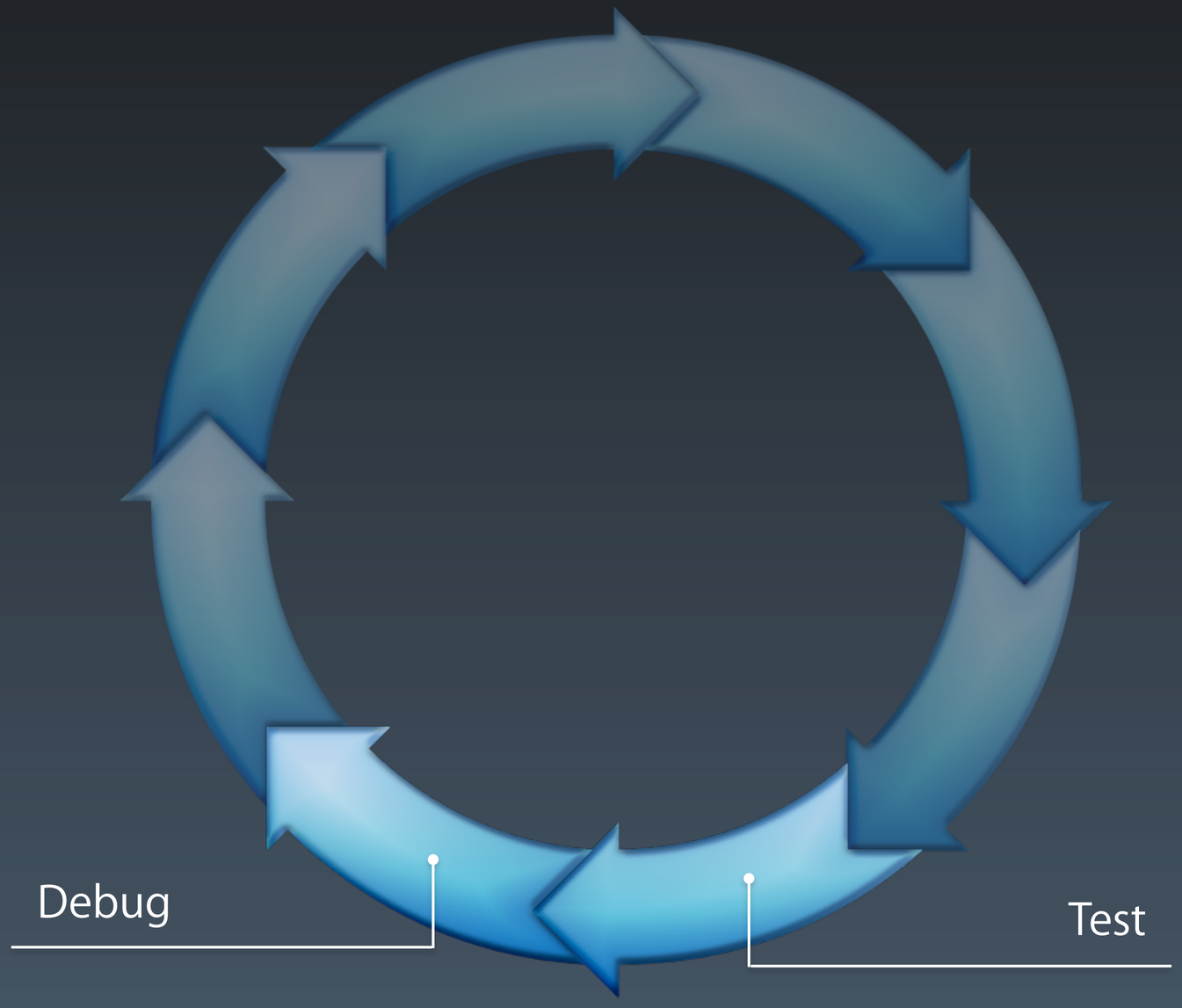
The State of Your Project

“Active” menu items and toolbar popups



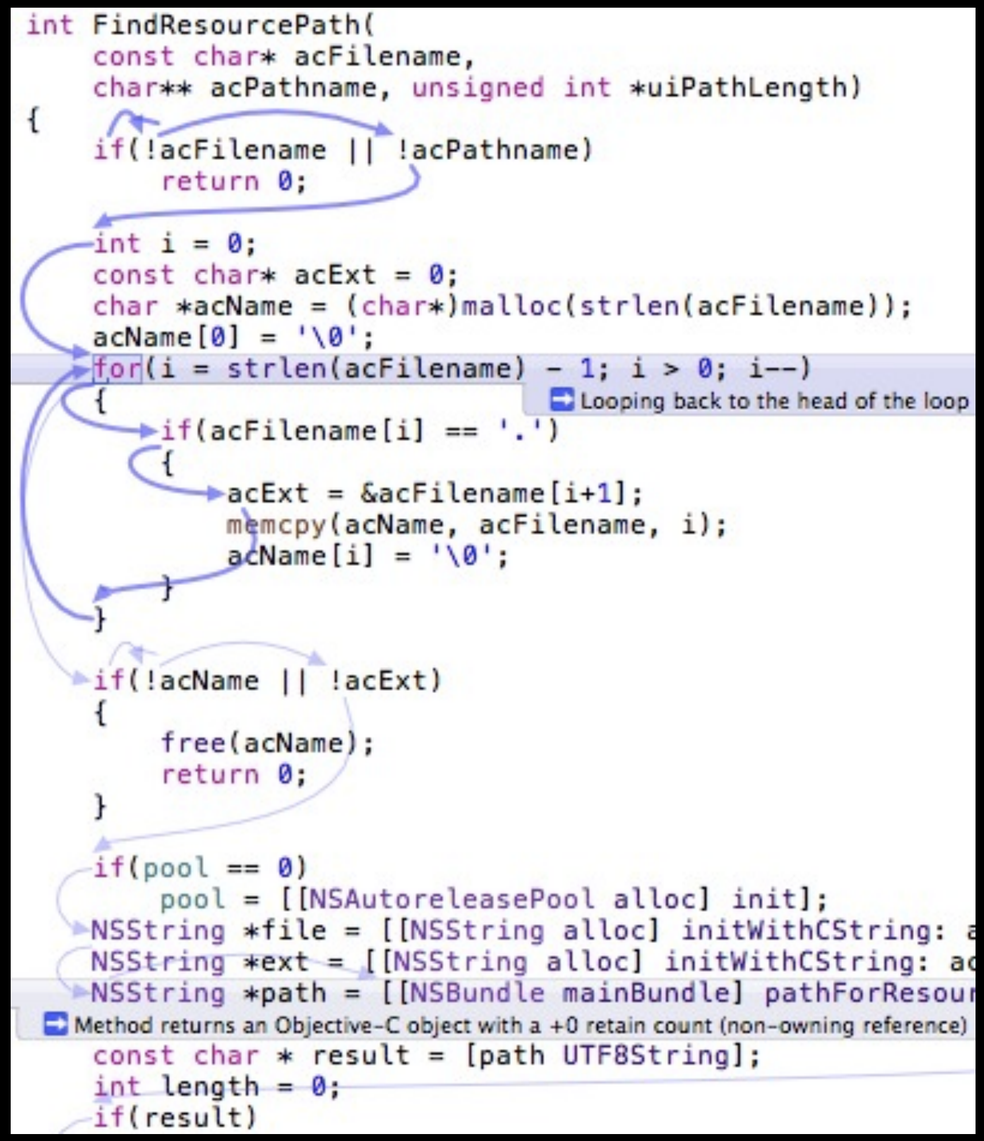
Mac OS X 10.5 SDK

Test and Debug



Stages of Testing

- Static analysis
- Unit tests
- Debugging your debug configuration
- Debugging your release configuration
- Evaluating your application in the field



Static Analysis

A compiler's-eye view of your project

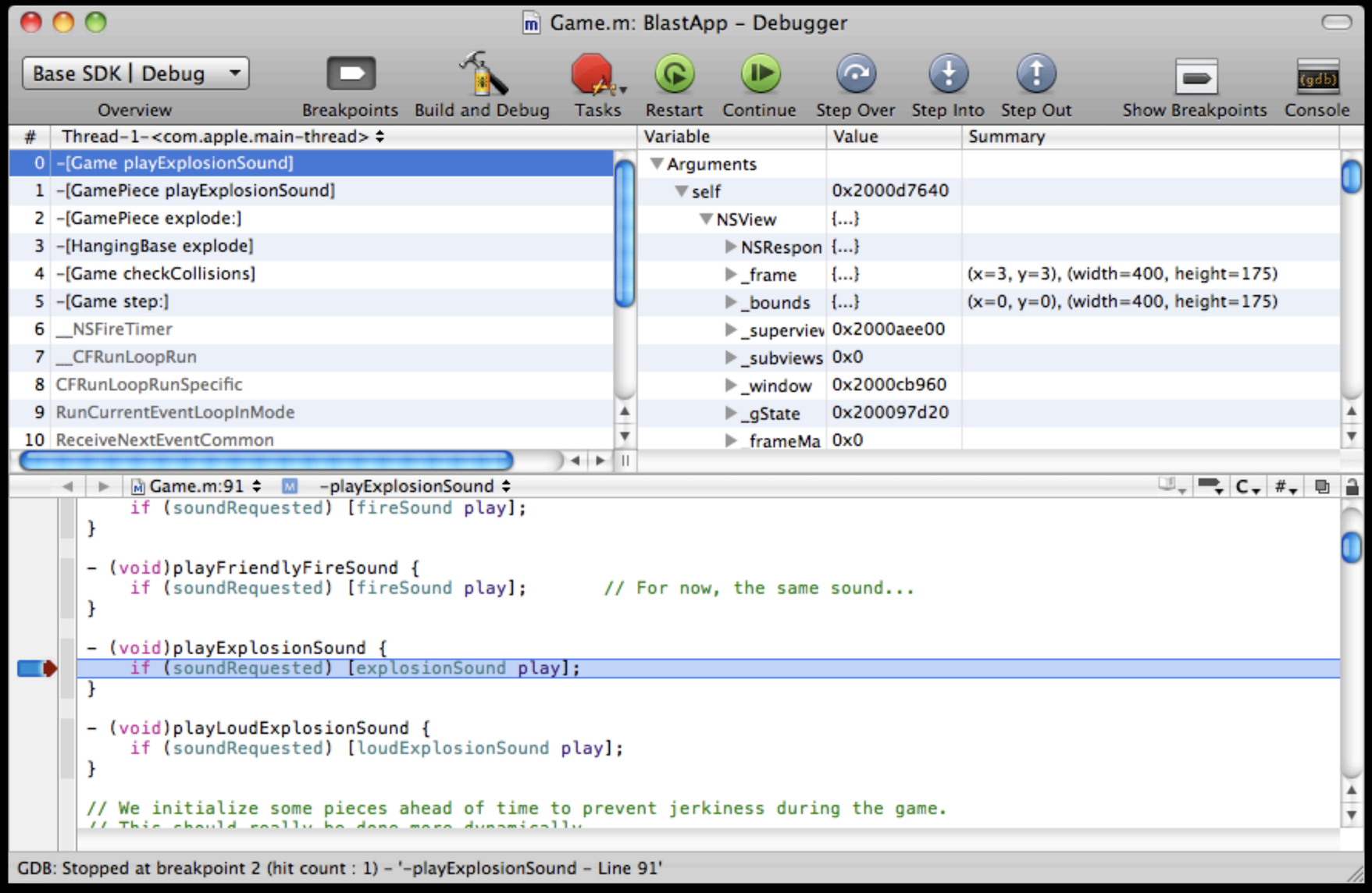
- Time consuming; you may only do this occasionally
- Finds logic flaws, dead code, leaks, and bad patterns
- Some risk of false positives

Unit Testing

- Software that tests software
- Xcode provides a framework that injects test cases into your code's process space and executes it
- You just write test case subclasses that call your code and return a result

Debugging

Interactive visual inspection and manipulation of your app



Demo

Building, analyzing, testing

Rick Ballard
Developer Tools Engineer

Summary

- Xcode is the hub of fully integrated development
- Mac OS X and iPhone development are substantially similar
- There's a lot to learn at the start
- Xcode probably has what you're looking for…

though it may be in a different place under a different name

Where to Go from Here

- Good documentation starting points
  - Xcode Overview
  - Mac OS X System Overview
  - iPhone OS System Overview
- Xcode's Welcome Panel
- Lots of great sessions at WWDC
  - Developer Tools sessions
  - Operating System sessions
- Visit us in the Lab!

More Information

Michael Jurewitz
Developer Tools Evangelist
jurewitz@apple.com

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc/attendee/

Related Sessions

Developer Tools State of the Union	Presidio Monday, 3:30PM
Introduction to iPhone Development Tools	North Beach Tuesday, 9:00AM
Compiler Overview	Marina Tuesday, 3:30PM
Customizing Xcode for Your Development Workflow	Pacific Heights Wednesday, 5:00PM

Labs

Xcode Tools Lab	iPhone Lab C Tuesday, 2:00PM
Xcode Tools Lab	iPhone Lab A Wednesday 3:15PM
Xcode Tools Lab	iPhone Lab B Thursday 10:15AM
Xcode Tools Lab	OS X Lab A Friday 9:00AM

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09